Deep Learning Lab1 Report

Chien-Hsun, Lai

March 28, 2018

1. Introduction

In this lab, we compare three different CNN models:

- Vanilla CNN 20/56/110
- ResNet 20/56/110
- ResNet pre-act 20/56/110

Comparing each models for:

- Same depth different architecture. Check if ResNet really outperform Vanilla CNN.
- Same architecture different depth. Check how will the models perform if we add more layers to them.

2. Experiment Setups

a. Model Details

We use tensorflow to implement our models. Tensorboard to visualize training results. All files are inside the cifar10 folder.

- dataset.py: downlaod and extract cifar10
- helper.py: simple helper functions
- input_cifar10.py: pipeline input data.
- resnet.py: implement models according to argument. All layers are defined in the inference() function
- train.py: train single model, main loop for training
- train.sh: train all the models, assign different parameters
- eval.py: evaluate model.

b. Hyper-parameters

Hyper-parameter are same for all the models except the number of layers.

- Optimizer: SGD+Momentum
- Batch size: 128
- Epochs: 164
- Learning rate: 0.1, divide by 10 at 81, 122 epoch
- Momentum: 0.9

- Weight decay: 0.0001
- Weight initialization: MSRA initialization
- Loss function: Cross-Entropy

3. Results

The following number are reported in the order of $depth = 20/56/110$

a. Vanilla CNN

Color: 20(Orange) / 56(Blue) / 110(Red)
Final Test Error: 9.910 / 15.290 / 87.410

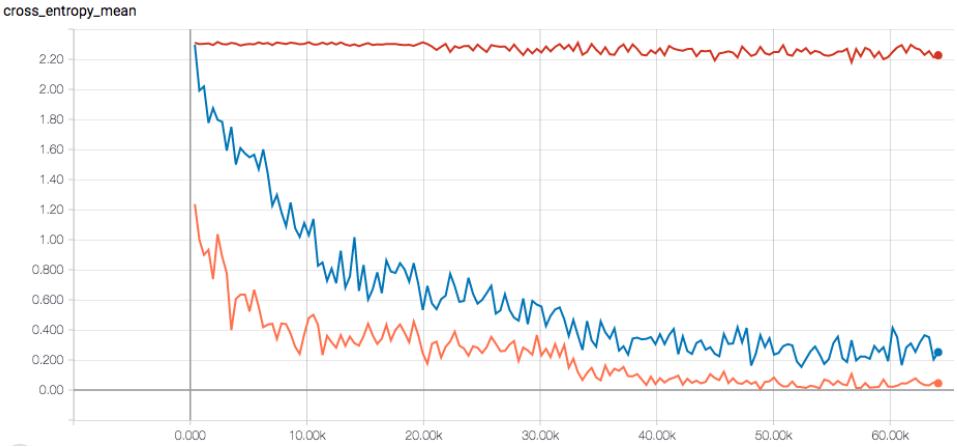

Figure 1: Vanilla CNN Training Loss.

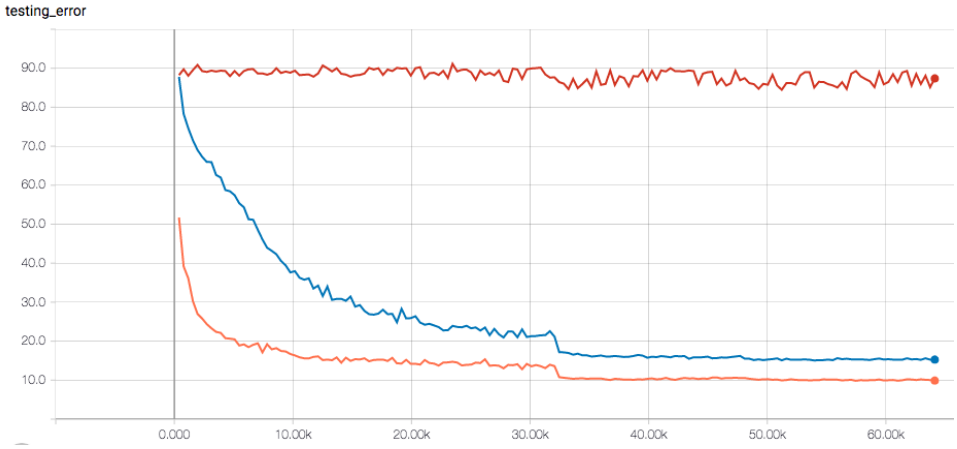

Figure 2: Vanilla CNN Test Error %.

b. ResNet

Color: 20(Blue) / 56(Red) / 110(Green)
Final Test Error: 9.190 / 7.400 / 7.460

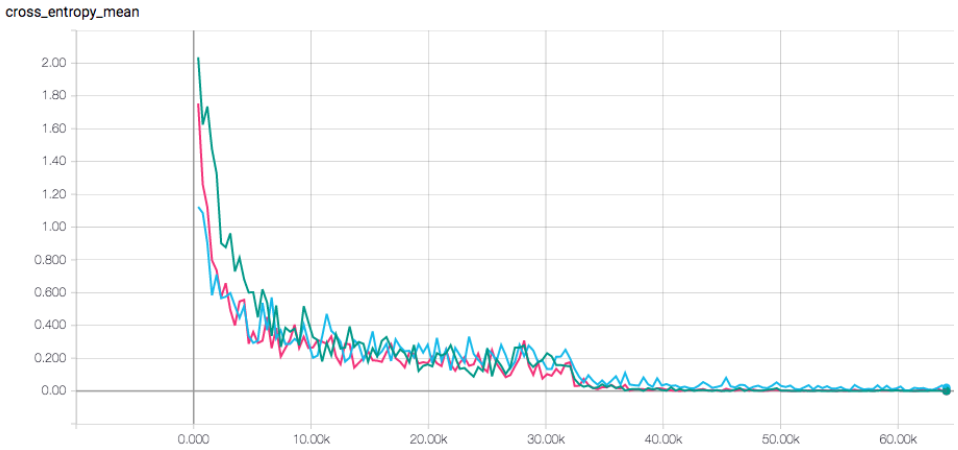

Figure 3: ResNet Training Loss.

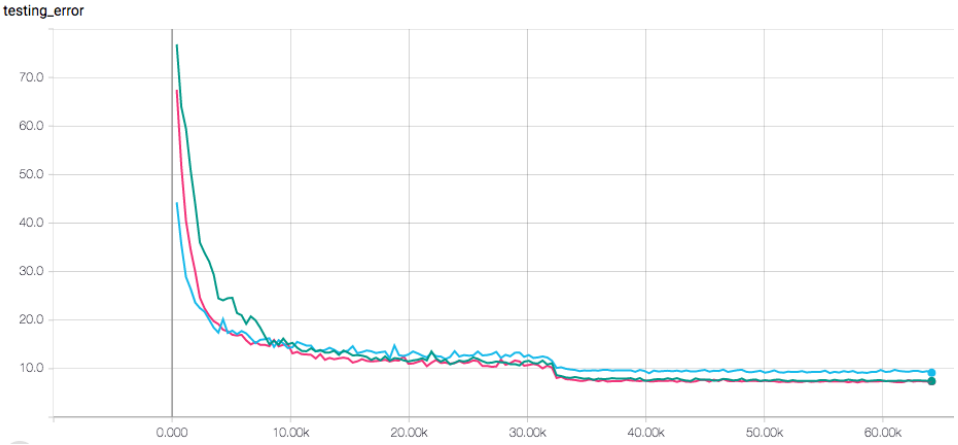

Figure 4: ResNet Test Error %.

c. Bonus: ResNet pre-act

Color: 20(Grey) / 56(Orange) / 110(Blue)
Final Test Error: 9.310 / 7.660 / 6.970

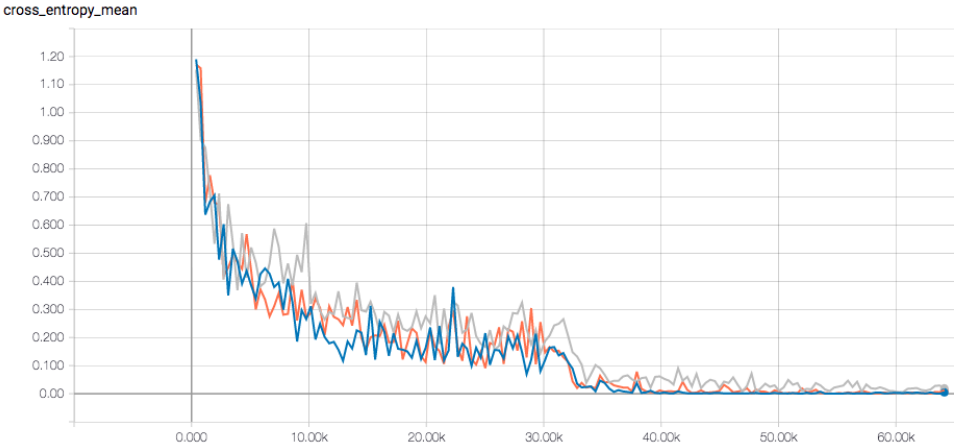

Figure 5: ResNet pre-act Training Loss.

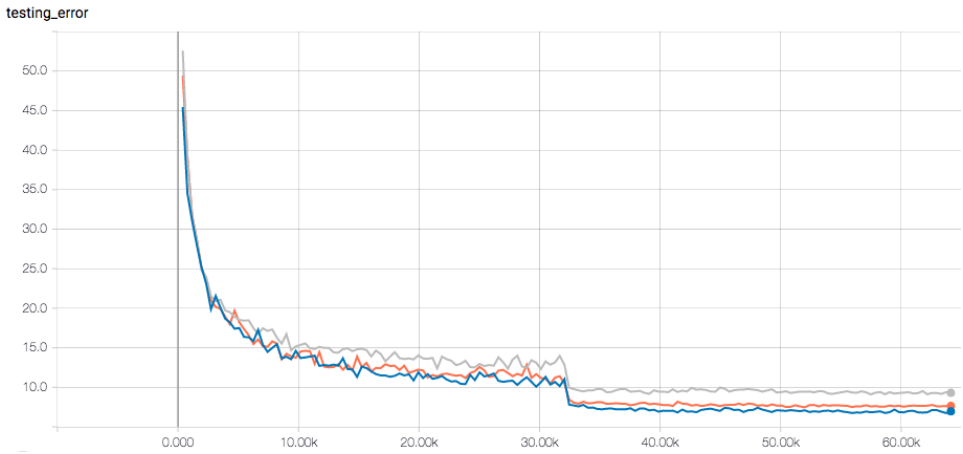

Figure 6: ResNet pre-act Test Error %.

d. Depth = 20

Color: Vanilla CNN(Orange) / ResNet(Blue) / ResNet pre-act (Grey)
Final Test Error: 9.910 / 9.190 / 9.310

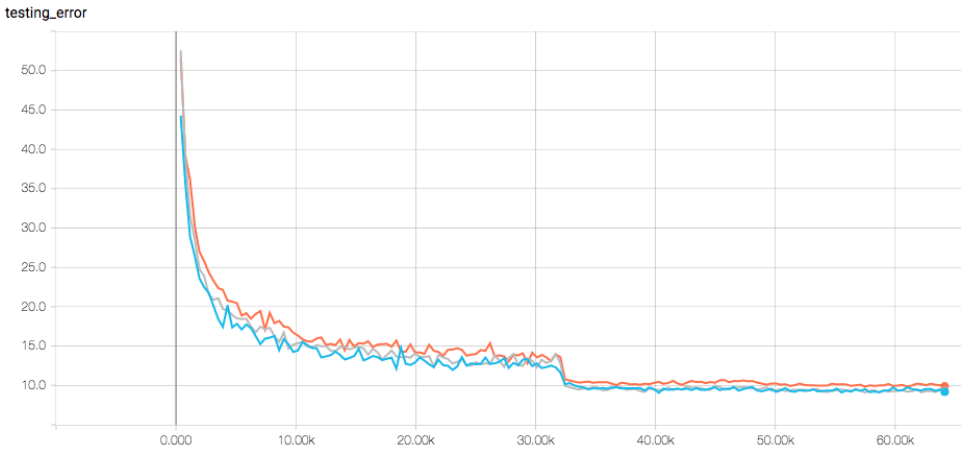

Figure 7: Test Error (Depth = 20) %.

e. Depth = 56

Color: Vanilla CNN(Blue) / ResNet(Pink) / ResNet pre-act (Orange)
Final Test Error: 15.290 / 7.400 / 7.660

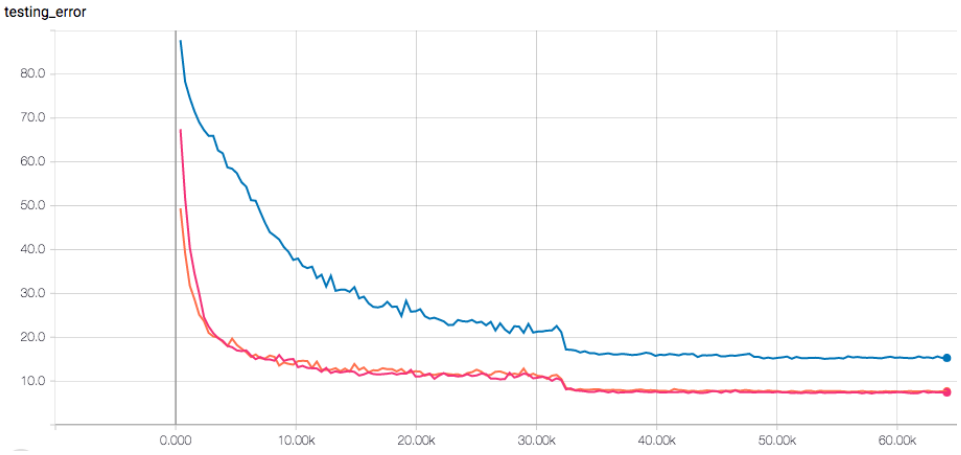

Figure 8: Test Error (Depth = 56) %.

f. Depth = 110

Color: Vanilla CNN(Red) / ResNet(Green) / ResNet pre-act (Blue)
Final Test Error: 87.410 / 7.460 / 6.970

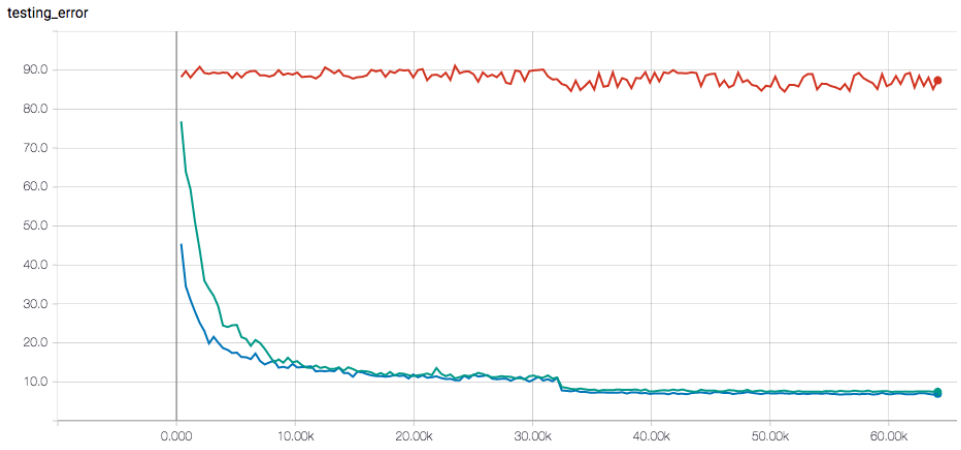

Figure 9: Test Error (Depth = 110) %.

4. Discussion

From the result, we can observe that as more and more layers are added. Vanilla CNN didn't perfrom better. While ResNet and ResNet pre-act all perform better or at least roughly the same as the layer increase. The gradient in ResNet can propagate back more easily. Because there are short cuts between the layers. ResNet pre-act performs slightly better than ResNet. Because the input can forward to any layer in ResNet pre-act. While in ResNet every two layers it passes through a ReLU. Which might cause some of the input signal being blocked.

ResNet pre-act 110 outperform all the other models in this experiment. This kind of architecture enables us to train arbitrary deep CNN.

Appendices

Source code for experiments
`https://github.com/jxcodetw/NCTU_DeepLearning/tree/master/lab1`